A pipeline based on Optical Character Recognition (OCR) and Image Caption (IC) to produce accessible books for the visually impaired

Xiang Wang
Department of Artificial intelligence
Peking University
2100013146@stu.pku.edu.cn

Xin Hao
Department of Artificial intelligence
Peking University
2100013152@stu.pku.edu.cn

Abstract

The growing spiritual and cultural requirements of the blind need to be met. However, barrier-free cultural products take a long time to prepare and are concentrated in more developed cities, they cannot be widespread to those in need. Here we show the pipeline to translate books for the visually impaired mainly with OCR and IC techniques. Our work demonstrates how to solve this problem and our result can build the foundation of this task. We anticipate our pipeline to be a starting point of research in producing accessible books for the visually impaired and we hope we can bring light to the blind.

1 Introduction

Nowadays, most accessible books are manually produced. Not only is this process inefficient, but it may bring many manual mistakes to these books. With the development of deep learning, many powerful techniques appear, including OCR and IC. OCR can recognize characters and IC can describe images. With these techniques, we can design a pipeline to produce accessible books automatically. The existing OCR technique is nearly perfect for this task, supporting up to 80 languages with high accuracy. But the existing IC technique is not capable because most of the IC models can only work with English and the descriptions are quite simple, while we need to describe more details of an image for the blind. Therefore, we need to produce a dataset and train IC models to get better performance.

2 Related Works

Optical Character Recognition OCR detects and recognizes characters in an image. Text detection algorithms include DB[17], EAST[39], SAST[33], etc. Text Recognition Algorithms include NRTR[25], RARE[27], SRN[38], etc. With the development of deep learning, the performance of OCR becomes better and better. Up to 80 kinds of languages are supported, and accuracy can be guaranteed in most situations. But because of lacking data with rare words, it will make mistakes when encountering some rare characters.

Image Caption IC is an automatic way to generate natural language descriptions for the given image. The early image caption way can be divided into two different categories. One is that Retrieval based image captioning likeFarhadi et al. [8], Hodosh et al. [10] andOrdonez et al. [23]. The other is Template based image captioning, such as Yang et al. [35], Mitchell et al. [21], and Ushiku et al. [29]. With the development of deep learning, Deep neural network-based image captioning become the mainstream. Though there are many different architectures, the most basic and widely-used model is an encoder-decoder model, which is introduced by Kiros et al. [12] from the machine translation field. By using the attention mechanism and other techniques, image caption can get both accurate

and fluent output in English now. But the Cross-Lingual Image Captioning and image captioning for non-realistic style pictures still need further research.

3 Methods

3.1 Overview

It consists of four parts: OCR, Image Extractor, IC, and text Typeset. Before we feed inputs into the pipeline, we need to preprocess the PDF file, turning every page of the PDF file into an image with the same height[1]. We call these images as page-images. OCR module[2] takes in page images, and output text boxes, which contain characters and corresponding positions(coordinates). Image Extractor also takes in page images, finds out where the pictures are, and cuts out pictures from origin pages. With text boxes and positions of pictures, we can typeset the characters into a half-completed book with picture labels in the proper place. IC module gets pictures from the Image Extractor and outputs descriptions of these pictures. Finally, we replace the picture labels with picture descriptions one by one and output the translated book as a .txt file. Fig. 1 demonstrates the whole pipeline.

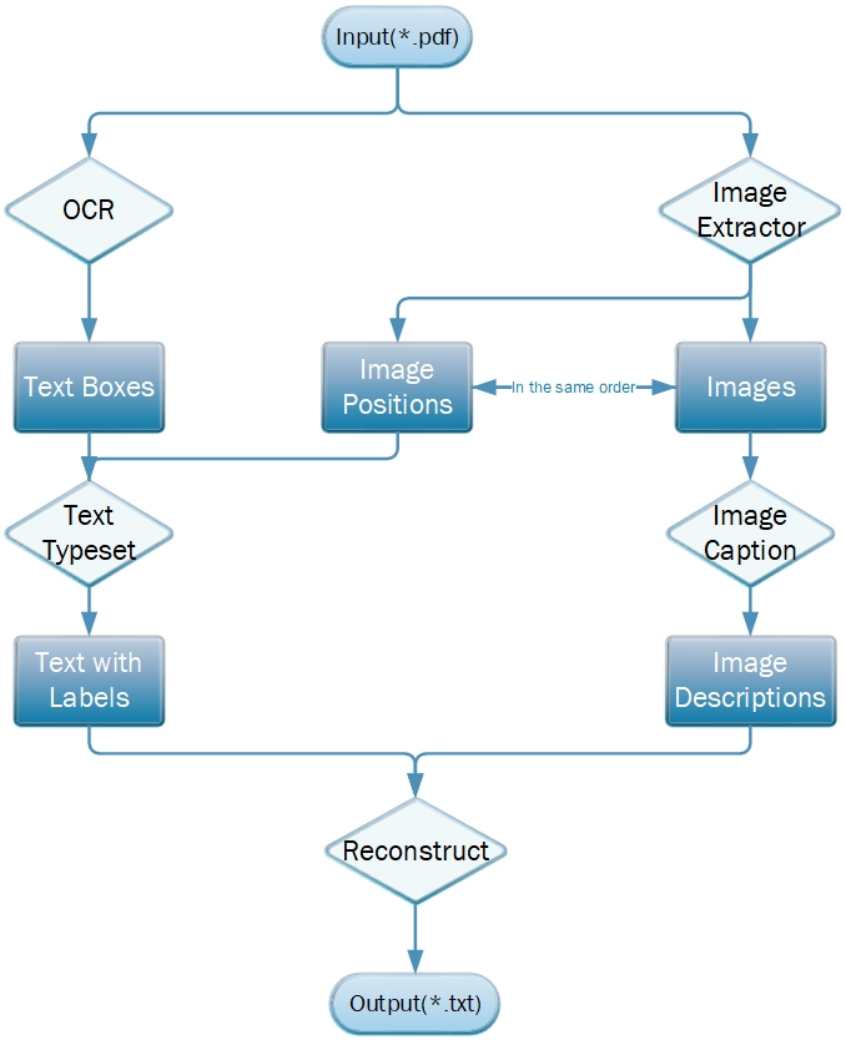


Figure 1: Pipeline

[1]Just the same height because the aspect ratio of one page varies from different books. If we turn pages to the same size, it will stretch or shrink the page and interfere with subsequent operations.

[2]Considering OCR technology is relatively mature, we choose to use the existing OCR module. After trying different OCR modules, we choose PaddleOCR because its API is easy to use and its performance is wonderful.

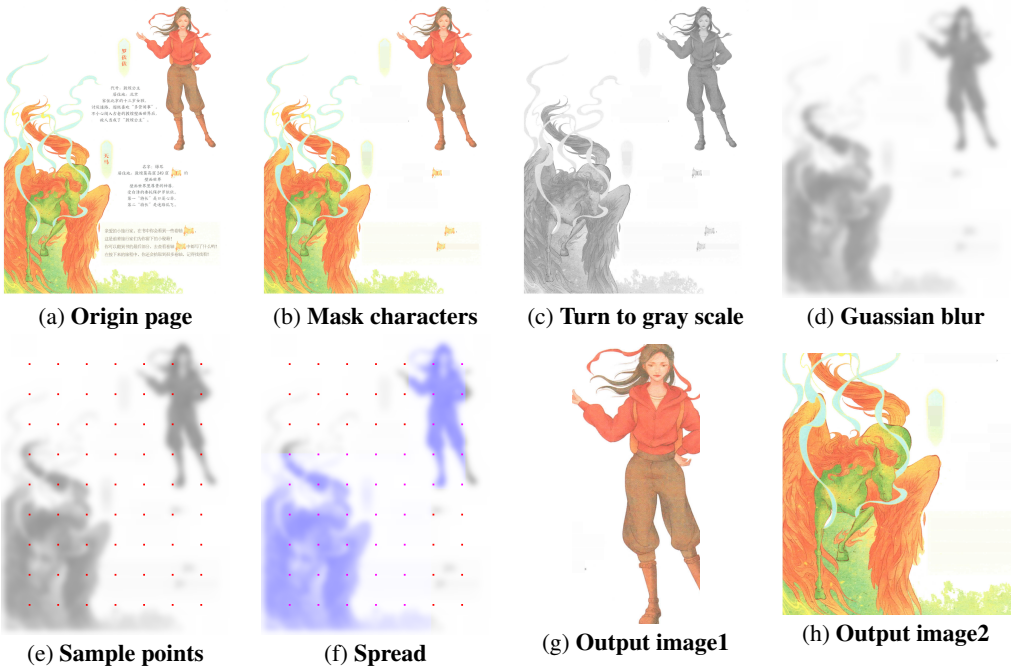
3.2 Image Extractor

This module needs to frame every picture on a page with a rectangular area[3] , and for a single page, it should output pictures separately and accurately, so that the IC module can produce a description for every picture. Of course, we can handle this problem with deep learning, but the traditional method can work well.

3.2.1 Motivation

We name our algorithm for Image Extractor as Sample-Spread. As its name, the major parts of this algorithm are sample and spread. We grid every page, and the grid points are the sample points. For every sample point, if it is an image-point[4] and we didn't visit this picture, then we call the spread function at this point. Otherwise, we ignore it and continue traversing sample points. The spread function is recursive. For an image point, we traverse its neighbors. For every neighbor, if it is also an image point and we didn't visit this point before, we call the spread function at this point recursively. When spreading, we keep updating area information with the coordinates of spread points. Because the spreading will not end until we arrive at the boundary of this picture, we can get an accurate rectangular area to cover the whole picture after exiting the spread function. Moreover, we can make sure there is only one picture in the area because spreading will end at the boundary. Then we continue traversing sample points until we visited all sample points. Finally, we cut out pictures from the original page with the area information we got from the spread function.

3.2.2 Implementation Details



(a) **Origin page** (b) **Mask characters** (c) **Turn to gray scale** (d) **Guassian blur**

(e) **Sample points** (f) **Spread** (g) **Output image1** (h) **Output image2**

Figure 2: **An example of how Image Extractor works**

The origin pages are RGB images, operations are slower than gray-scale images. Considering the input book file can contain thousands of pages, the algorithm should be as fast as possible. So, we'd better turn pages into gray-scale before the beginning of Sample-Spread.

For simplification, we consider a point as an image point when the pixel is non-white. But colorful characters on the page sometimes will interfere with our judgment of whether a point is an

[3]Many pictures in books are irregular, and we can still get a precise boundary of pictures with the spread function. But IC requires a rectangular area, so we use a rectangular area rather than a precise boundary in this module.

[4]If a point is inside a picture, we call it an image point.

image point. With the text boxes output by OCR module, we can locate these characters and mask them out.

The spread function is recursive, but python limits recursive depth (default maximum 1000). When we traverse neighbors, if we just visited pixel by pixel, it is easy to go beyond maximum depth. On one hand, we need to set the recursive depth limitation as a bigger number. On the other hand, even if there isn't such a limitation, suppose that the whole page is a picture and its size is 2000×1000, we need to traverse 2×10^6 points for a single page. That's too slow and it is unacceptable. As a result, we adopt the stride parameter. When we spread to a point at (x, y), we traverse one of its neighbors at $(x + stride, y)$ rather than $(x + 1, y)$. With a proper stride parameter, not only can the algorithm be quite faster, but we can still get accurate area information.

However, stride brings an unavoidable problem: truncation. The boundary of a picture will not be as precise as we traverse pixel by pixel, sometimes the edge of the picture will be cut out. That happens when we traverse with a stride that is larger than the distance between that point and the boundary. Choosing a fixed stride parameter to suit every traverse operation is impossible. So we introduce adapted stride-traverse. We introduce a random parameter $delta$ when traversing neighbors. That is $(x + stride + delta, y)$ rather than $(x + stride, y)$. If we go beyond the boundary(neighbor is not an image point), we try to find a neighbor with stride=stride/2. In this way, we can mitigate the truncation problems.

When we grid the page, parameter grid size (distance between two nearby sample points) is important. If it's too large, we may miss some pictures. But if it's too small, the algorithm will be slow. We ended choose grid size as width/9 and height/11, which means we sample 63 points every page, 7 points in direction of width, and 9 points in direction of height.

Sometimes we may find the algorithm extracts the icon at the edge of the page as an image. It's normal for this algorithm but we don't hope that happens. So, we set a threshold for the picture size, only if the size of the area is bigger than 1/200 of the page area, can we consider it as a picture.

After we handle all of these problems, we can get an image extractor with great performance.

3.3 Image Caption

To complete the image captioning, we have two basic choices. One is to use an image-to-Chinese network, while the other is a two-step strategy with image-to-English as the first step and English-to-Chinese as the second. To choose a better strategy, we tried a lot of networks, a sample is as follows.

Model	Output
ClIP[24]-Chinese	一个美丽的油画看上去是一个灰色的树
GIT[32]	digital art selected for the #
BLIP[15]	an image of an oriental man and bird

Table 1: **Different output of the same sample**

What we can see in this sample is that with the pre-trained model, BLIP reserve the most details in the picture, which is just what we want. The same difference happened in almost every other sample image. A possible reason is that there are much fewer models trained in Chinese and there are not so many datasets providing Chinese captions, which causes the Chinese datasets available to be often noisy. These two cause it hard for image-to-Chinese models to get better results. Among English models, though GIT got higher CIDEr[30] scores, with the pre-trained model, BLIP performs better. Also, BLIP has a more friendly API, which may reduce our work to add it to our final program. As a result, we finally choose BLIP as our baseline in image Captioning

3.3.1 BLIP

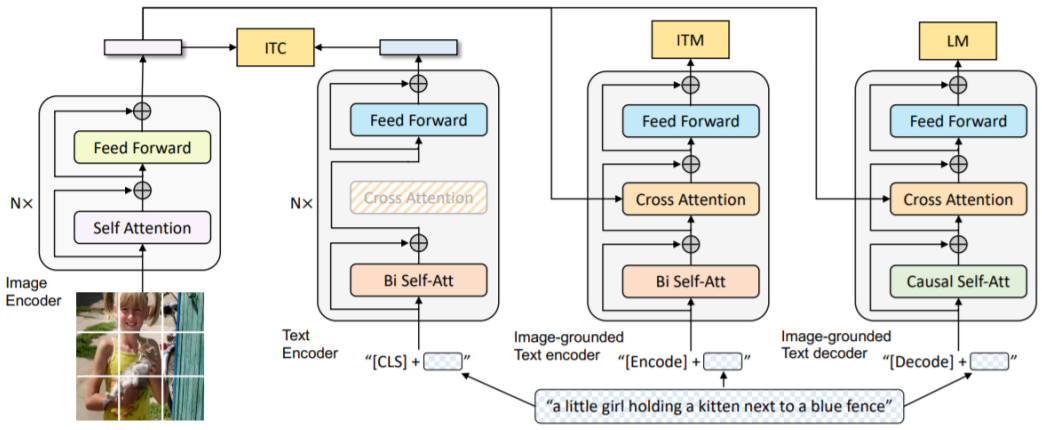


Figure 3: **Architecture of BLIP** from Li et al. [15]

BLIP is a unified model for VLP tasks that allows us to train from noisy image-text pairs. For its multi-task application, the authors introduce a Multimodal mixture of Encoder-Decoder (MED). In contrast to CLIP, its architecture is much more complex. CLIP uses ViT or ResNet as an image encoder and Transformer as a text encoder. By aligning the image features and text features, it can generate a caption from a given image. But as 3 shows,BLIP has four main module:

- **Image Encoder (ViT)**, used to extract the image features, which is the basis of all three tasks.
- **Text Encoder (BERT)**, with Image-Text Contrastive Loss (ITC) as objective, used to align feature space of Image Encoder Transformer and Text Encoder Transformer.
- **Image-grounded Text Encoder (Modified BERT)**, with an additional Cross Attention module inserted between Bi Self-Att and Feed Forward, using Image-Text Matching Loss (ITM) as an objective to predict negative or positive. It is used as the multimodal representation of the image-text pair, to adjust the Fine-grained alignment.
- **Image-grounded Text Decoder (Modified BERT)**, replacing Bi Self-Att in Image-grounded Text Encoder with Causal Self-Att, using Language Modeling Loss (LM) as an objective to generate caption.

Another reason why BLIP perform better than other model is its unique CapFilt module. This module solves the problem that many image-text pairs from the Web are too noisy for training. In this module, a captioner is used to generate a caption for a given Web image. It is an Image-grounded Text Decoder to decode the text with a given image. After that, a Filter is used to remove the noise in mage-text pairs. It uses an Image-grounded Text Encoder to judge whether the image matches the text, and then filters the noisy text and improves the quality of the dataset.

In short, BLIP provides us with a simple way to make the dataset and train the model, and it works well with its pre-trained model on our sample tasks.

3.3.2 Translation

Because there is no Chinese-based BLIP model, a translator is needed for the final program. A basic idea is to use the free API of Baidu, Google, Youdao, etc. After considering the cost, accuracy,

and fluency, we finally choose the Baidu translation API. sers By easily registering a free account, users can simply input their app-id and app-key to our program and finish the objective task.

4 Conclusion

In this paper, We demonstrate how to combine existing techniques to design a pipeline for producing accessible books. We have pointed out deficiencies of these techniques when applying them to the pipeline, and how to solve these problems. As an important part of the pipeline, We designed an efficient Image Extractor with traditional methods and got great performance. Although the development of deep learning is relatively mature, considering there are few applications in this field, our work can be a starting point. Further works can be covering more rare characters in OCR, finding a more robust method of the typeset module, designing a better model for IC to get more detailed descriptions, etc.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[2] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4715–4723, 2019.

[3] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 71–79, 2018.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[7] Desmond Elliott, Stella Frank, and Eva Hasler. Multilingual image description with neural sequence models. *arXiv preprint arXiv:1510.04709*, 2015.

[8] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010. 1

[9] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

[10] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47: 853–899, 2013. 1

[11] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision*, pages 2407–2415, 2015.

[12] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1

[13] Weiyu Lan, Xirong Li, and Jianfeng Dong. Fluency-guided cross-lingual image captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1549–1557, 2017.

[14] Rémi Lebret, Pedro Pinheiro, and Ronan Collobert. Phrase-based image captioning. In *International Conference on Machine Learning*, pages 2085–2094. PMLR, 2015.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 4, 5

[16] Xirong Li, Weiyu Lan, Jianfeng Dong, and Hailong Liu. Adding chinese captions to images. In *Proceedings of the 2016 ACM on international conference on multimedia retrieval*, pages 271–275, 2016.

[17] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020. 1

[18] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: A spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016.

[19] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017.

[20] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE international conference on computer vision*, pages 2623–2631, 2015.

[21] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alexander Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756, 2012. 1

[22] Takashi Miyazaki and Nobuyuki Shimizu. Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790, 2016.

[23] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011. 1

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 4

[25] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition, 2019. 1

[26] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[27] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016. 1

[28] Yipeng Sun, Jiaming Liu, Wei Liu, Junyu Han, Errui Ding, and Jingtuo Liu. Chinese street view text: Large-scale chinese text reading with partially supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9086–9095, 2019.

[29] Yoshitaka Ushiku, Masataka Yamaguchi, Yusuke Mukuta, and Tatsuya Harada. Common subspace for model and similarity: Phrase learning for caption generation from images. In *Proceedings of the IEEE international conference on computer vision*, pages 2668–2676, 2015. 1

[30] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 5

[31] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016.

[32] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 4

[33] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. A single-shot arbitrarily-shaped text detector based on context attended multi-task learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1277–1285, 2019. 1

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[35] Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 444–454, 2011. 1

[36] Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Russ R Salakhutdinov. Review networks for caption generation. *Advances in neural information processing systems*, 29, 2016.

[37] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[38] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020. 1

[39] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 1

[40] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. More grounded image captioning by distilling image-text matching model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4777–4786, 2020.